The NeOn Ontology Engineering Toolkit

Peter Haase, Holger Lewen, Rudi Studer, Thanh Tran
Institut AIFB, Universität Karlsruhe (TH), Germany
{haase,lewen,studer,tran}
@aifb.uni-karlsruhe.de

Michael Erdmann
Ontoprise GmbH
Karlsruhe, Germany
erdmann
@ontoprise.de

Mathieu d'Aquin
Enrico Motta
Knowledge Media Institute
Open University, UK
{m.daquin, e.motta}
@open.ac.uk

1. MOTIVATION

Ontologies are a key technology enabling semantic interoperability and integration of data and processes. We are now entering a phase of knowledge system development, in which ontologies are produced in larger numbers and exhibit greater complexity. Since more and more content becomes available online, there is an increasing need for technologies that enable the reuse of existing (Semantic) Web resources from within the ontology engineering environment itself.

In the NeOn project[1] we aim at advancing the state of the art in using ontologies for large-scale semantic applications. This is realized by providing an infrastructure for networked ontology management and engineering capable of suiting the community's needs [1]. The heart of this infrastructure is the NeOn Toolkit for engineering contextualized networked ontologies and semantic applications.

The shift from closed semantic applications, characterized by a monolithic corporate ontology, to open semantic applications, characterized by networks of ontologies, implies that a number of other aspects like dealing with context, collaboration or data and web integration become crucial. The NeOn Toolkit was built with the web-centric elements of semantic technologies in mind: much like Web 2.0 environments emphasize distributed content production, the NeOn Toolkit features methods and tools for managing knowledge that is distributed, heterogeneous, contextualized, and developed collaboratively.

2. OVERVIEW OF THE NEON TOOLKIT

The NeOn Toolkit is a new generation ontology engineering environment, combining industrial-strength robustness with a comprehensive support for the ontology engineering life-cycle. The target community for the platform includes both researchers from the semantic web, knowledge management and related areas, as well as professional users from commerce and industry. The core of the Toolkit is freely available in open source[2] and provides the reference implementation of the NeOn architecture [3].

The NeOn Toolkit is designed around an open and modular architecture, which includes infrastructure services, such as a registry and a repository, and supports distributed components for ontology management, reasoning and collaboration in networked environments. Strong emphasis is put on networked ontology management, i.e. support for engineering ontologies that are embedded in a network of ontologies via rich semantic relationships, including models for modular ontologies and mappings across ontologies [2]. Based on Eclipse, the Toolkit defines an open framework for plugin developers.

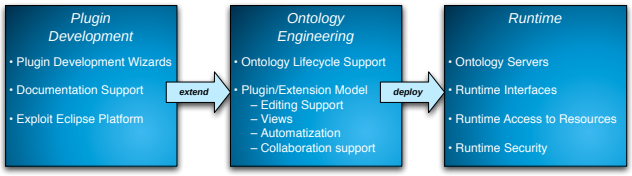

Figure 1: The main focus of the NeOn the ontology engineering process lies on lifecycle aspects (center box). The required functionality is provided by an initial set of plugins, which is expected to grow through contributions by the community. A set of developer tools supports the plugin-creation process (box on the left). Finally users of the toolkit will be offered a runtime platform which serves as the base for applications using semantic technologies (box on the right).

A typical NeOn Toolkit plugin consists of a set of Eclipse-plugins that encapsulate particular ontology engineering functionalities, interacting with other plugins to support the entire ontology lifecycle. A central datamodel plugin encapsulating basic ontology management functionalities is the entry point (and a minimal requirement) for every plugin. Many plugins, such as the initial set of the toolkit, offer extension mechanisms for other plugins. This makes the platform very flexible, as it does not only support predefined extensions that fit into certain slots, but also extensions of extensions. The NeOn Toolkit also supports the integration of distributed plugins, including for example web service based plugins to access remote components.

Already within the NeOn community a number of plugins addressing different ontology-lifecycle activities have been released or are under development. These include support for visual modeling (OntoModel), ontology reuse (Watson), ontology learning (Text2Onto), ontology mapping (R2O, FOAM), ontology diagnosis and repair (RaDON), as well as an ontology registry (Oyster). In addition to sup-

[1] http://www.neon-project.org/
[2] http://neon-toolkit.org/

WWW2008, April 21–25, 2008, Beijing, China.

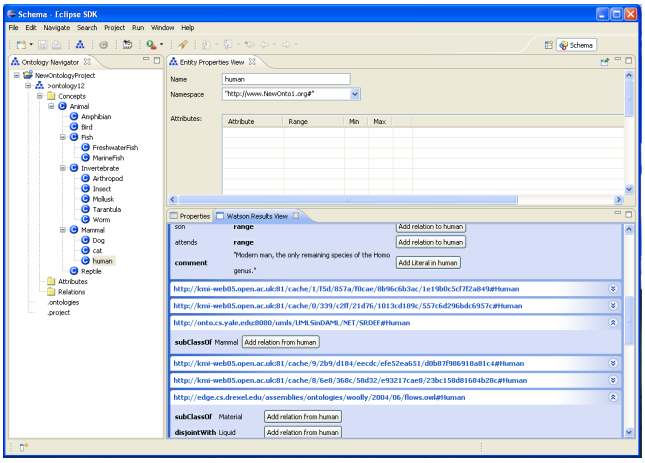

Figure 2: The NeOn Toolkit with the Watson Plugin

porting an open, collaborative approach to developing functionalities, the extensible architecture of the NeOn Toolkit is also crucial to enable us to realize certain key advanced functionalities, such as integrating external infrastructures like DBMS, Web services and other (Semantic) Web resources, supporting a dual language approach—this includes both OWL and rules, and ensuring compliance with Service-Oriented Architectures (SOA).

The ontology engineering sector is rapidly becoming a strategic area for many companies. What sets the NeOn Toolkit apart from other ontology engineering environments is the combination of a comprehensive set of state-of-the-art functionalities for ontology engineering with solutions that facilitate the integration of conventional software, as well as Web technologies, in semantic applications. The integration is achieved by means of the open source framework. The first open source version of the Toolkit has been unveiled at the 2007 International Semantic Web Conference and, as befits an open, collaborative enterprise, we welcome contributions from both users and developers, who are keen to help us test and develop the Toolkit.

3. SHOWCASE EXAMPLE: THE WATSON PLUGIN

In the presentation, we will demonstrate the NeOn Toolkit and its functionalities. Moreover, we will demonstrate how to extend the functionalities of the ontology engineering environment by developing NeOn plugins to support additional lifecycle activities. In particular, we will demonstrate how to reuse and integrate online Semantic Web resources into the ontology engineering process, using the Watson plugin as a specific example.

Watson[3] is a gateway to the Semantic Web that provides an efficient access point to online ontologies and semantic data. As such, it plays three main roles: (1) it collects the available semantic content on the Web, (2) analyzes it to extract useful metadata and indexes, and (3) implements efficient query facilities to access the data.

The Watson plugin[4] for knowledge reuse allows the ontology engineer to query Watson and reuse results from within the NeOn Toolkit. To do this the engineer simply selects an entity in the current ontology and triggers the Watson search in order to find existing descriptions, which can then be reused in (i.e. integrated into) the currently edited ontology.

For that purpose, it provides interface elements that seamlessly integrate with the graphical user interface of the NeOn toolkit, by implementing extensions of standard Eclipse widget elements (views, menus, etc.). It also takes benefit from features provided by Eclipse to manage and store properties and settings, which enables the plugin to integrate properly within the overall environment. Through these interface elements, the Watson plugin allows the user to select entities of the currently edited ontology he/she would like to inspect, and to automatically trigger queries to Watson, as a remote Web service. Results of these queries, i.e. semantic descriptions of the selected entities in online ontologies, are displayed in an additional view allowing further interactions.

Figure 2 provides an example, where the user has selected the concept "human" and triggered a Watson search. The view on the right provides the query results (a list of definitions of class human which have been found on the Semantic Web) and allows easy integration of the results by simply clicking on one of the different "add"-buttons.

Finally, the core of the plugin is the component that interacts with the core of the NeOn toolkit: its datamodel. Statements retrieved by Watson from external ontologies can be integrated in the edited ontology, requiring for the plugin to extend this ontology through the NeOn toolkit datamodel and data management component. Finally, thanks to the common infrastructure provided by the NeOn toolkit, interactions with other plugins can be envisaged. Different plugins support different activities in the lifecycle of ontologies, and one possible interaction could be the use of the mapping plugin for keeping track of the external resources included by the Watson plugin.

The demonstration illustrates two key aspects of the NeOn toolkit in terms of development: (1) the plugin development process relying on a common, flexible architecture allowing to seamlessly integrate new features in the toolkit environment, and (2) the ability to flexibly integrate external Web resources and services into the ontology lifecycle management process.

[3] `http://watson.kmi.open.ac.uk`

[4] `http://watson.kmi.open.ac.uk/editor_plugins.html`

4. REFERENCES

[1] M. Dzbor, E. Motta, R. Studer, Y. Sure, P. Haase, A. Gómez-Pérez, R. Benjamins, and W. Waterfeld. Neon - lifecycle support for networked ontologies. In *Proceedings of 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies (EWIMT-2005)*, pages 451–452, London, UK, NOV 2005. IEE.

[2] P. Haase, S. Rudolph, Y. Wang, S. Brockmans, R. Palma, J. Euzenat, and M. d'Aquin. D1.1.1 networked ontology model. Technical Report D1.1.1, Universität Karlsruhe, NOV 2006.

[3] T. Tran, P. Haase, H. Lewen, Ó. Muñoz-García, A. Gómez-Pérez, and R. Studer. Lifecycle-support in architectures for ontology-based information systems. In *Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, pages 508–522, 2007.